

Язык программирования Pascal

Линейные алгоритмы

Выражение

- **Выражение** – совокупность операндов и операций.
- **Операнды** – константы, переменные, функции.
- **Арифметические операции** –
 - бинарные операции $+$ $-$ $*$ $/$, (* - умножение / - деление)
 - унарные операции $+$ и $-$.
 - бинарные операции **div** и **mod** определены только для целых чисел, переменных типа **integer**.
 - **div** – деление нацело $5 \text{ div } 2 = 2$
 - **mod** – остаток от деления $43 \text{ mod } 10 = 3$

Приоритет операций

Приоритет - порядок выполнения операций в выражении.

Операции, имеющие одинаковый приоритет, выполняются слева направо.

Для изменения порядка операций используют **скобки ()**, скобки обладают наивысшим приоритетом.

Таблица приоритетов операций

вычисление функций	0
not	1 (высший)
*, /, div, mod, and,	2
+, -, or, xor	3
=, <>, <, >, <=, >=, in	4 (низший)

Правила записи выражений

Все записи в одну строку.

Знаки умножения опускать нельзя.

**Главный критерий правильности записи выражения
- совпадение порядка выполнения операций с
принятым в математике.**

$$\sqrt{5(1+a)} \rightarrow \text{sqrt}(5 * (1 + A))$$

$$(1+x)^2 \rightarrow \text{sqr}(1+x)$$

$$|a+bx| \rightarrow \text{abs}(a + b * x)$$

Арифметические выражения

Какие из приведенных выражений правильно записаны на языке ПАСКАЛЬ?

- 1 $(A+B+C)/2$ ← 1 3 7 9
- 2 A^2C
- 3 $SIN(X+2)*2$ ←
- 4 $2COS(A+2)-A$
- 5 $X^Y+Z-5.1$
- 6 $(X+Y*3.2-A)B$
- 7 $X+Y/(Z*Z)$ ←
- 8 $X*X+Y*Y=R*R$
- 9 $SQRT(A*2+B*B)$ ←
-

Оператор присваивания :=

Выполняет изменение значения переменной

Синтаксис

<имя переменной> := <выражение>;

Семантика

Вычисляется **<выражение>** в правой части,

Результат записывается в **<переменную>** .

Ограничение. Тип выражения должен быть совместим по присваиванию с переменной.

Например:

Одинаковые типы совместимы.

Выражение типа **integer** можно присвоить переменной типа **real**.

Обратное неверно.

Пример использования оператора присваивания.

a := (3 + 5) * 8;

b := a + 2;

Оператор присваивания

Среди приведенных выражений указать правильные операторы присваивания

1 **$X := X + 5$**

5 **$X + 2 := Y$**

2 **$Y := 7$**

6 **$K + 5 := X + Y$**

3 **$WRITE("C=", C:6:2)$**

7 **$WRITE("A+B")$**

4 **$Z := SQR(Z) + 5.3 * X$**

8 **$X := X * X + A * X + B$**

1 2 4 8

Оператор присваивания

Какое значение получит переменная Y после выполнения следующей программы?

$X := 5;$

$A := 2;$

$B := -1;$

$Y := A * X + B;$

9

Оператор присваивания

Какое значение получит переменная Y после выполнения следующей программы?

A:=12;

B:=14;

A:=10;

B:=B+5;

Y:=A+B;

END.

Оператор присваивания

Какое значение получит переменная X после выполнения следующей программы?

A:= 3;

B:= 4;

A:=(A+B)/2+A*2;

X:=SQRT(A+B*B-0.5);

END.

Оператор присваивания

Какое значение получит переменная S после выполнения программы?

A:=5;

B:=6;

S:=A*B;

A:=-1;

B:=3;

S:=S+A*B

Оператор ввода read, readln

Синтаксис

read(<список переменных>);

readln(<список переменных>);

Семантика

- Происходит считывание данных с клавиатуры и запись их в переменные из <списка переменных> по порядку.
- Вводить данные нужно через пробел или по нажатию <Enter>.
- Программа продолжится, когда будут считаны все данные.
- **Readln** – вызывает переход к следующей строке.

С процедурой ввода связан ряд **ошибок** (например, если должно быть получено целое число, а вводится 'ABC').

Пример: **read (a,b)**
 readln (x,y,z)

Вывод write, writeln.

Вывод в окно вывода

Синтаксис:

Write (<список выражений>);

writeln(<список выражений>);

Параметры в списке перечисляются через запятую.

Семантика:

1. Вычисление значений выражений.
2. Вывод значений.

Процедура **writeln** после вывода своих параметров осуществляет переход на следующую строку.

Пустой **writeln** вызывает пропуск строки.

Вывод **write writeln**.

Форматы вывода.

В процедурах вывода **write** и **writeln** можно указать *формат вывода*, в виде **:m:n**, где m и n - целые значения. .

:m - *ширина поля вывода*.

:n - *количество знаков после десятичной точки*

Если *длина выводимого значения меньше* ширины поля вывода, то выводимый текст *слева* дополняется пробелами. Выравнивание по правому краю.

Если *длина выводимого значения больше* ширины поля вывода, то формат игнорируется.

Значения с форматом вывода вида **:m** будут представлены в экспоненциальной форме.

Например, если a, b - целые переменные, то при выполнении операторов

```
a:=-2437; b:=13555;  
writeln(a:6, 'Привет!':9);  
writeln(b:1);
```

в окно вывода будет выведен следующий текст:

```
-2437  Привет!  
13555
```

Например:

```
writeln(-14.859:10:3); //  ___-14.859  
writeln(-14.859:10:5); //  _-14.85900  
writeln(-14.859:10:2); //  ____-14.86  
writeln(-14.859:10:0); //  _____-15  
writeln(-14.859:10:7); //  -14.8590000  
writeln((0,1):10:1); //  _ (0.0,1.0)  
(здесь символом _ изображены пробелы).
```

Оператор write, writeln

- | | |
|---------------------------------------|---|
| 1 <code>write (' A=',A:5:2)</code> | 5 <code>write(x:6:3,f:7:4)</code> |
| 2 <code>write(a,b,c)</code> | 6 <code>write(s,d)</code> |
| 3 <code>writeln</code> | 7 <code>write('s= ', s:6:3, ' ',w=',w:8:4)</code> |
| 4 <code>write (' x=', x: 10:5)</code> | 8 <code>write(sqrt(25)+5)</code> |

- Какие из операторов вывода, приведенные в таблице, выводят только значения переменных? **2 5 6**
- Какие из приведенных операторов вывода выводят имена переменных и их значения? **1 4 7**
- Какой оператор определяет пропуск строки?

Примеры использования :=

Пример 1. Перемена местами двух целых значений. Дано: x, y;

```
Program swap1;  
var x, y, v: integer;  
begin  
  read(x,y);  
  writeln('x=',x, ' y=', y);  
  v := x;  
  x := y;  
  y := v;  
  writeln('x=',x, ' y=', y);  
  end.
```

```
Program swap2;  
var x, y: integer;  
begin  
  read(x, y);  
  writeln('x=',x, ' y=', y);  
  x := x + y;  
  y := x - y;  
  x := x - y;  
  writeln('x=',x, ' y=', y);  
  end.
```


Задания на линейные алгоритмы

Разработать математическую модель. Составить блок-схему алгоритма и тест. Написать программу на языке Pascal.

- **Задача 1.** Даны катеты прямоугольного треугольника **a** и **b**. Найти его гипотенузу **c**, периметр **P** и площадь **S**. **(begin 12)**
- **Задача 2.** Найти длину окружности **L** и площадь круга **S** заданного радиуса **R**. ($L=2\pi R$; $S= \pi R^2$; считать, что $\pi=3,14$). **(begin 7)**
- **Задача 3.** Скорость первого автомобиля **V1** км/ч, второго — **V2** км /ч, расстояние между ними **S** км. Определить расстояние между ними через **T** часов, если автомобили удаляются друг от друга. **(begin 36)**
- **Задача 4. (begin 20)** Найти расстояние между двумя точками с заданными координатами (x_1, y_1) и (x_2, y_2) на плоскости. Расстояние вычисляется по формуле
$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$
- **Задача 5. (begin 21)** Даны координаты трех вершин треугольника: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Найти его периметр **P** и площадь **S**. Для нахождения площади треугольника со сторонами **a**, **b**, **c** использовать формулу Герона

$$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)},$$

где $p = (a + b + c)/2$ — полупериметр

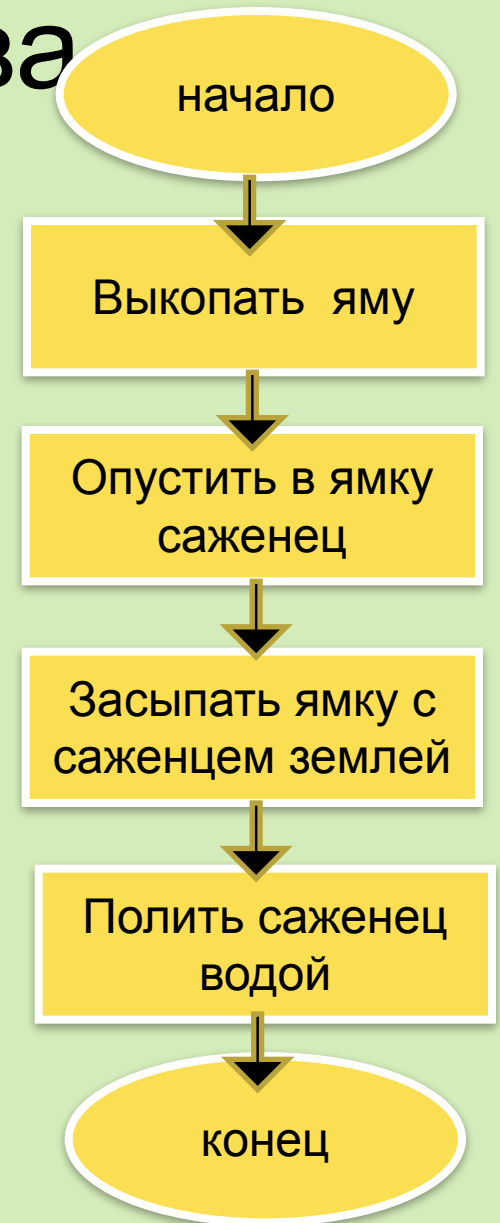
73. Написать программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Например, число 12.5 должно быть преобразовано к виду 12 руб. 50 коп.

Операции **div** и **mod**

- **ВНИМАНИЕ!!!** Операции **div** и **mod** определены только для данных типа integer
- Целочисленное деление $x \text{ div } y = x / y$, округленное до ближайшего целого по направлению к нулю.
- Остаток от целочисленного деления $x \text{ mod } y = x - (x \text{ div } y) * y$.
- Пример использования
 1. Для определения **четности** числа:
 $x \text{ mod } 2 = 0 \leftrightarrow x$ — четное $x \text{ mod } 2 \neq 0 \leftrightarrow x$ — нечетное
 2. Для операций с цифрами числа
Сумма цифр целого трехзначного числа A.
 $s := a \text{ div } 100 + a \text{ mod } 100 \text{ div } 10 + a \text{ mod } 10;$

Задание : составьте алгоритм посадки дерева

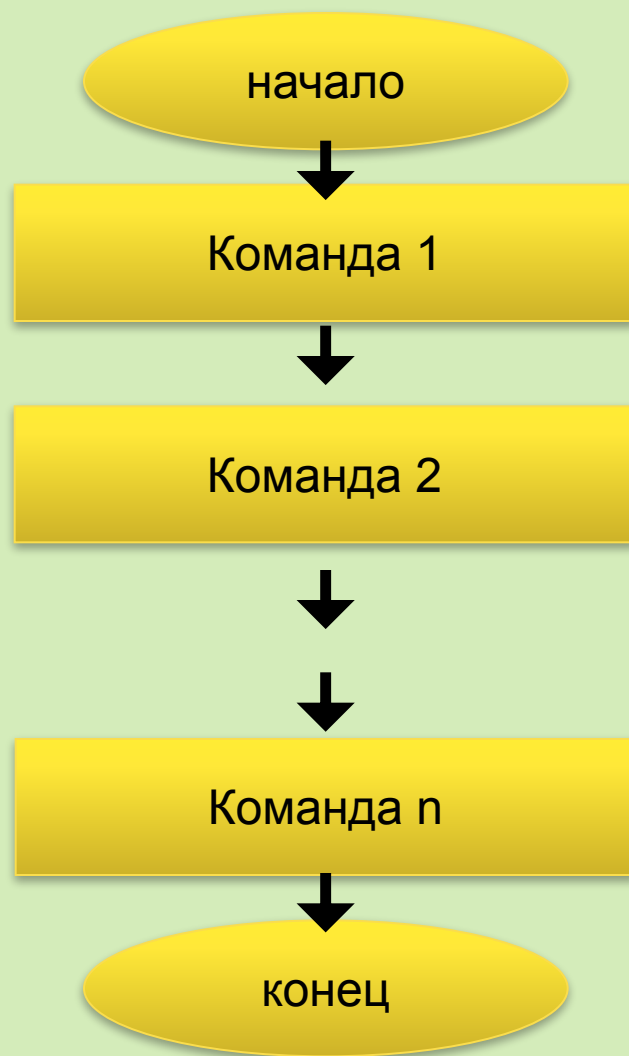
1. Выкопать в земле ямку;
2. Опустить в ямку саженец;
3. Засыпать ямку с саженцем землей;
4. Полить саженец водой.



Линейный Алгоритм

- **Алгоритм, в котором команды выполняются последовательно одна за другой, называется линейным алгоритмом**
- (Пример: Алгоритм посадки дерева.)

Базовая структура линейного алгоритма

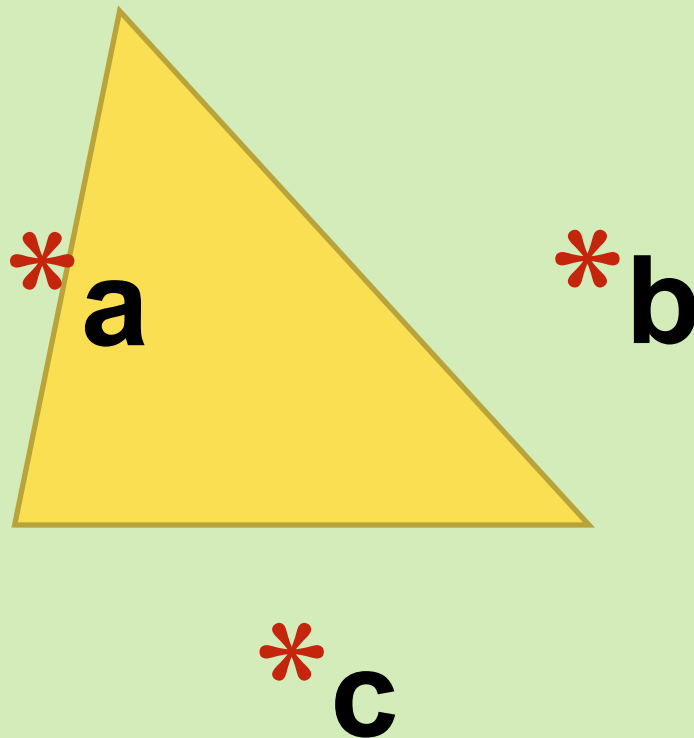


Этапы решения задач на компьютере:

1. Постановка задачи .
2. Построение математической модели.
3. Алгоритмизация.
4. Написание задачи на языке программирования.
5. Отладка и тестирование программы.
6. Анализ полученных результатов.

Задача

- Вычислить периметр произвольного треугольника по его трем сторонам.



Решение

- 1. Постановка задачи

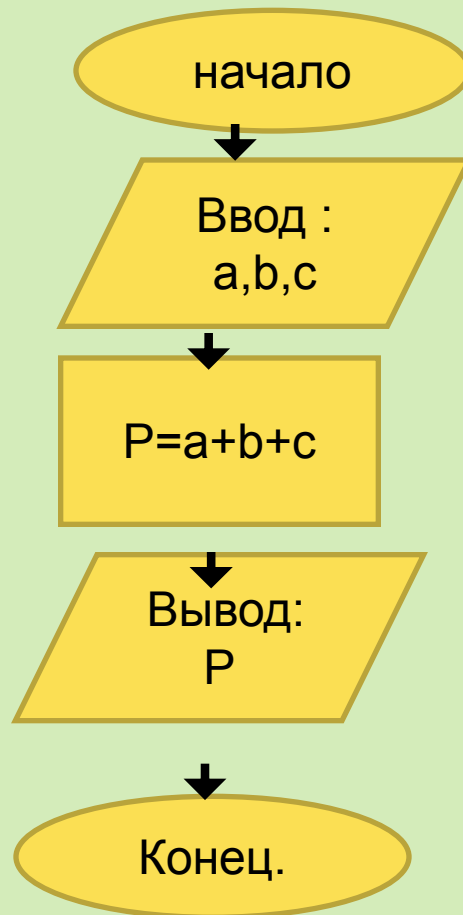
* Исходные
данные

* Результат

Переменная	Смысловое значение	Тип значения
a	Стороны треугольника	вещественный
b		вещественный
c		вещественный
P	Периметр треугольник	вещественный

Решение

- 2. Построение математической модели:
- $P = a + b + c$
- 3. Алгоритм решения:



Решение

- 4.Программа:
- Program zad_1;
- Uses crt;
- Var a,b,c:integer;
- Begin
- Clrscr;
- Write('Введите значение сторон
треугольника');
- Readln(a,b,c);
- P:=a+b+c;
- Write('Периметр треугольника P=', P);
- Readln;
- End.

- 5.Тест.

Задаем значения
$a=3$
$b=5$
$c=8$
$P=3+5+8=16$

Что происходит после запуска программы?

Введите через запятую значение сторон треугольника 3,5,8 <ENTER>

- После этого в переменные заносятся введенные значения
- $a = 3$ $b = 5$ $d = 8$
- Выполняется оператор присваивания
- $P = a + b + c$
- В переменную P заносится значение 16 ($P = 16$)
- Выносится результат на экран
Периметр треугольника $P = 16$

Домашнее задание

- 1. Выучить записи в тетради.
- 2. Задача: Вычислить периметр произвольного прямоугольника по двум его сторонам. (Запись в тетради блок-схемы и алгоритм программы на языке ABC Pascal)